

WHAT IS CLAIMED IS:

1. A method of distributing a distributed application on a network,
comprising:
disposing said distributed application on at least one server within said network,
5 said distributed application comprising a plurality of modules;
transferring a first portion of said plurality of modules to a client device via a first
communication channel;
attempting to establish a second communication channel between said client device
and said server; and
10 transferring a second portion of said distributed application to said client device
from said server when said second communication channel can not be established.
2. The method of Claim 1, further comprising:
attempting to establish said second communication channel after said second
portion of said distributed application has been transferred; and
15 transferring a third portion of said distributed application to said client device when
said second communication channel can not be established after said second portion has
been transferred.
3. The method of Claim 1, wherein said network comprises a cable network,
and the act of transferring said first portion comprises downloading a client portion to a
20 client device.
4. A distributed application for use on a cable network, comprising:
at least one server portion disposed on at least one of a plurality of servers of said
network, said at least one server portion comprising a plurality of modules;
at least one client portion disposed on at least one of a plurality of client devices of
25 said network, said at least one client portion comprising at least one module;
at least one mutable module, said mutable module being adapted to run on either at
least one of said plurality of servers or at least one of said plurality of client devices;
wherein said at least one mutable module is adapted to maintain at least one
network partition.
- 30 5. The distributed application of Claim 4, wherein the location of said at least

one mutable module is determined at least in part by the resources of said at least one client device.

6. The distributed application of Claim 4, further comprising at least one peer module having an interface, said interface preventing said peer module from being moved from said at least one server to said at least one client device.

7. The distributed application of Claim 6, wherein said interface comprises an application programming interface (API).

8. A message protocol useful for communicating between various entities of a distributed application on a client-server network; comprising:

an address object having a first address associated therewith, said first address comprising a local address of at least a first of said entities of said distributed application, said address object being adapted for communication between said at least first entity and a second entity, said at least first and second entities being disposed on the same server.

9. The message protocol of Claim 8, wherein said address object further comprises a second address, said second address comprising a network address of at least a third entity of said distributed application, said at least third entity being disposed on a server different than that of said at least first and said second entities.

10. The protocol of Claim 9, wherein at least one of said first and second addresses comprise a queue identifier.

11. The protocol of Claim 9, wherein said network address comprises an Internet Protocol (IP)/User Datagram Protocol (UDP) port pair.

12. The protocol of Claim 9, wherein said network address comprises a virtual address that maps to an Internet Protocol (IP)/ User Datagram Protocol (UDP) port pair.

13. A distributed application architecture for use with a cable network having a plurality of servers and a plurality of client devices, comprising:

a distributed application comprising at least one server portion and at least one client portion; said at least one server portion being disposed on at least one of said servers, said at least one client portion being disposed on at least one of said client devices, said at least one client device being in communication with said at least one server;

a runtime environment comprising a plurality of processes, said processes facilitating the allocation of client device load between servers on said network;
and

5 a message protocol adapted for communication between individual ones of said processes of said runtime environment, and between said individual ones of said processes and said at least one server portion.

14. A distributed application architecture for use with a cable network having a plurality of servers and a plurality of client devices, comprising:

10 at least one distributed application comprising at least one server portion and at least one client portion; said at least one server portion being disposed on at least one of said servers and having a plurality of process threads running thereon, said at least one client portion being disposed on at least one of said client devices, said at least one client device being in communication with said at least one server and providing processing requests thereto;

15 a runtime environment comprising a plurality of processes, said processes being configured to distribute client device load between individual ones of said plurality of servers on said network, and between individual ones of said plurality of process threads associated with said at least one distributed application; and

20 a message protocol adapted for communication between individual ones of said processes of said runtime environment, and between said individual ones of said processes and said at least one server portion.

15. A cable network head-end system, comprising:

a plurality of servers;

25 a distributed application comprising a first server portion; said first server portion being disposed on at least one of said servers;

a runtime entity comprising a plurality of software processes, said processes facilitating the allocation of client device load between servers on said network;

30 a message protocol adapted for communication between individual ones of said software processes of said runtime environment, and between said individual

ones of said processes and said first server portion; and

a data network in data communication with at least a portion of said processes of said runtime entity and said servers.

5 16. The head-end system of Claim 15, further comprising a statistics process and at least one timing process, said statistics process and said at least one timing process cooperating to allocate client device load among said plurality of servers.

17. The head-end system of Claim 15, further comprising a second server portion associated with said runtime entity, said second server portion being in data communication with said first server portion via said message protocol.

10 18. A distributed application adapted for operation on a client-server network, comprising:

at least one server portion adapted for operation on at least one server of said network;

a message protocol configured for interprocess communication;

15 15 at least one client portion adapted for operation on at least one client device of said network, said at least one client portion being in data communication with said at least one server portion using at least said message protocol; and

20 at least one mutable module adapted for transmission from said at least one server to said at least one client device, said at least one mutable module comprising at least one vertical (client server) partition.

19. The distributed application of Claim 18, wherein said server portion and said client portion further comprise a plurality of modules having horizontal (peer) partitions.

25 20. A method of optimizing the performance of a cable network comprising a plurality of servers, comprising:

providing a distributed application comprising at least one server portion and at least one client portion, said at least one server portion being disposed on at least one of said servers and having at least one critical portion and at least one process thread running thereon;

30 receiving a plurality of processing requests from said at least one client portion at

said at least one server portion;

arbitrating access to said at least one critical portion of said at least one server portion by said processing requests;

measuring the value of at least one parameter associated with said access of said at
5 least one critical portion by said processing requests;

generating, based at least in part on said value, a new process thread; and
processing at least one of said plurality of requests using said new process thread.

21. The method of Claim 20, wherein the act of arbitrating access comprises
protecting said at least one critical portion using a wait-and-lock mechanism.

10 22. The method of Claim 21, wherein said wait-and-lock mechanism
comprises a queue.

23. The method of Claim 21, wherein said wait-and-lock mechanism
comprises a semaphore.

24. The method of Claim 20, wherein the act of measuring the value of at least
15 one parameter comprises measuring a time period beginning with initiation of at least one
of said processing requests.

25. A method of optimizing the performance of a cable network comprising a
plurality of servers and a plurality of client devices, comprising:

providing at least one distributed applications comprising at least one server
20 portion and at least one client portion, said at least one server portion being disposed on a
at least one server and having at least one critical portion and at least one process thread
running thereon;

receiving a plurality of processing requests from said at least one client portion at
said at least one server;

25 arbitrating access to said at least one critical portion by said plurality of processing
requests;

measuring the value of at least one parameter associated with said access of said at
least one critical portion by said processing requests;

distributing, based at least in part on said value, said at least one process thread to a
30 server of said network other than said at least one server; and

processing, at said other server, at least one of said plurality of requests using said at least one process thread.

26. The method of Claim 25, further comprising assigning to said at least one server portion and said at least one client portion an identifier so that said client portion may identify its corresponding server portion.

27. The method of Claim 26, further comprising generating a new process thread on said at least one server based at least in part on said value.

28. The method of Claim 27, wherein the act of arbitrating access comprises protecting said at least one critical portion using a wait-and-lock mechanism.

29. The method of Claim 28, wherein said wait-and-lock mechanism comprises a queue.

30. A method of balancing the load on a plurality of servers in a client-server network, comprising:

providing a distributed application, said distributed application a first server portion and at least one client portion, said first server portion having at least one critical portion associated therewith;

disposing said first server portion on a first of said plurality of servers;

receiving processing requests from said at least one client portion at said at least one critical portion of said first server portion;

obtaining data relating to the loading of individual ones of said servers resulting from said distributed application and other applications running on said network;

spawning a second server portion having at least one critical portion on a second of said plurality of servers, said second server being selected based at least in part on said data;

distributing at least a portion of said processing requests to said second server; and processing at least a portion of said processing requests using said second server portion.

31. A method of dynamically communicating between portions of a distributed application within a client-server network having a plurality of servers and a plurality of client devices, comprising:

providing a distributed application having at least one server portion and at least one client portion, said at least one server portion being disposed on at least one server within said network, and said client portion being disposed on at least one client device within said network;

5 assigning at least one virtual address to each of said at least one server portion and said at least one client portion, said at least one virtual address uniquely identifying the location of its respective server or client portion anywhere within said network;

moving said at least one server portion and/or said at least one client portion to another of said servers or client devices, respectively; and

10 maintaining communication between said client and server portions using at least said virtual address.

32. The method of Claim 31, further comprising:

storing records in at least one database, said records containing the mapping of said virtual addresses for said at least one server and client portions; and

15 updating said records in response to said act of moving.

33. The method of Claim 31, wherein said virtual addresses comprise Uniform Resource Identifiers (URI).

34. A cable network head-end system having a plurality of servers and a plurality of client devices, comprising:

20 at least one distributed application comprising at least one server portion and at least one client portion; said at least one server portion being disposed on at least one of said servers and having a plurality of process threads running thereon, said at least one client portion being disposed on at least one of said client devices, said at least one client device being in communication with said at least one server and providing processing requests thereto;

25 at least one service process running on at least one of said plurality of servers;

a runtime entity comprising a plurality of processes, said processes being configured to distribute client device load between individual ones of said plurality of servers on said network;

30 a message protocol adapted for communication between individual ones of said

37

processes of said runtime entity, and between said individual ones of said processes and said at least one server portion; and

a communications interface adapted for communication between said at least one server portion and said at least one service process.

5 35. The system of Claim 34, wherein said processes of said runtime entity are further configured to distribute client device load between individual ones of said plurality of process threads associated with said at least one distributed application.

36. The system of Claim 34, wherein said communications interface comprises OpenCable compliant local interface.

10 37. The system of Claim 34, wherein said at least one distributed application comprises at least one mutable module adapted for transmission from a server to a client device, said at least one mutable module comprising at least one vertical (client-server) partition.

15 38. A client-server network communications architecture used for communication between a plurality of server portions and a plurality of client portions associated with one or more distributed applications, said communication facilitating load-balancing of the network, comprising:

20 a statistical process running on at least one server of said network, said statistical process adapted to gather data relating to the use of resources within said at least one server; and

25 a first message protocol adapted to transmit said data from said statistical process to at least one of said first or second server portions, and between said first and second server portions.

30 39. The method of Claim 38, further comprising a second message protocol adapted for communication between respective ones of said server portions and said client portions.

40. The method of Claim 39, further comprising a third message protocol adapted for communication between respective ones of said client portions.

41. A method of configuring a client-server network having server resources and a plurality of client devices, and a distributed application adapted for use thereon,

comprising:

providing a first process running within said server resources, said first process configured to obtain data relating to the usage of said server resources;

receiving, at said server resources, a server resource request from a client device;

5 selecting a server within said server resources based at least in part on said data from said first process;

starting the server portion of said distributed application on said selected server;

and

10 downloading the client portion of said distributed application to said client device from said selected server.

42. The method of Claim 41, further comprising:

starting said client portion at said client device; and

43. sending a message to said server portion indicating that said client device has been started.

15 43. The method of Claim 41, further comprising providing a second process running within said server resources, said second process configured to perform the acts of selecting and downloading.

44. The method of Claim 41, wherein the act of downloading further comprises passing an identifier associated with said server portion to said client device, said identifier
20 facilitating communication between said server and client portions.

45. A method of operating a distributed application in a client/server network having a plurality of servers and at least one client device associated therewith, comprising:

selecting a client portion at said at least one client device;

25 executing said client portion;

requesting a server portion for said client portion from said plurality of servers;

selecting an appropriate server on which to operate said server portion;

running said server portion on the selected server;

informing said client portion that the server portion is available; and

30 running said distributed application based on input from a user at said client

device.